

# JobTrack - Final Project Writeup

---

## Team Details

---

**Team Name:** JobTrack

**Team Members:** Xavier Yin (jy963)

**Deployed Project URL:** <https://jobgenie-six.vercel.app>

**Source Code URL:** <https://github.com/Enfoirer/jobgenie>

## Project Idea

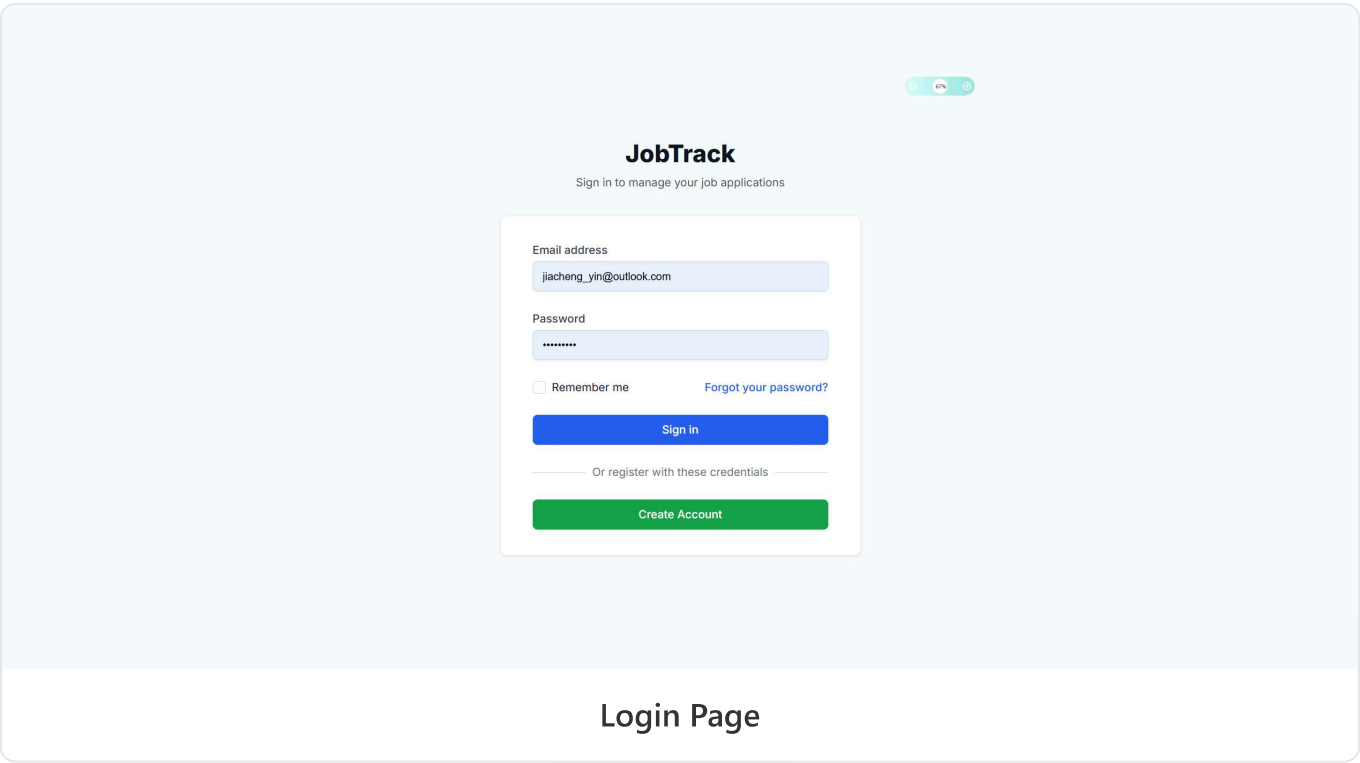
---

JobTrack is a comprehensive job application tracking system designed to help job seekers organize and monitor their job search process. The application allows users to track job applications across different stages (Applied, Interviewing, Offer, Rejected), upload and manage application-related documents, and view a detailed timeline of status changes for each job. The goal is to provide job seekers with a centralized platform to streamline their job search efforts and maintain better visibility into their application progress.

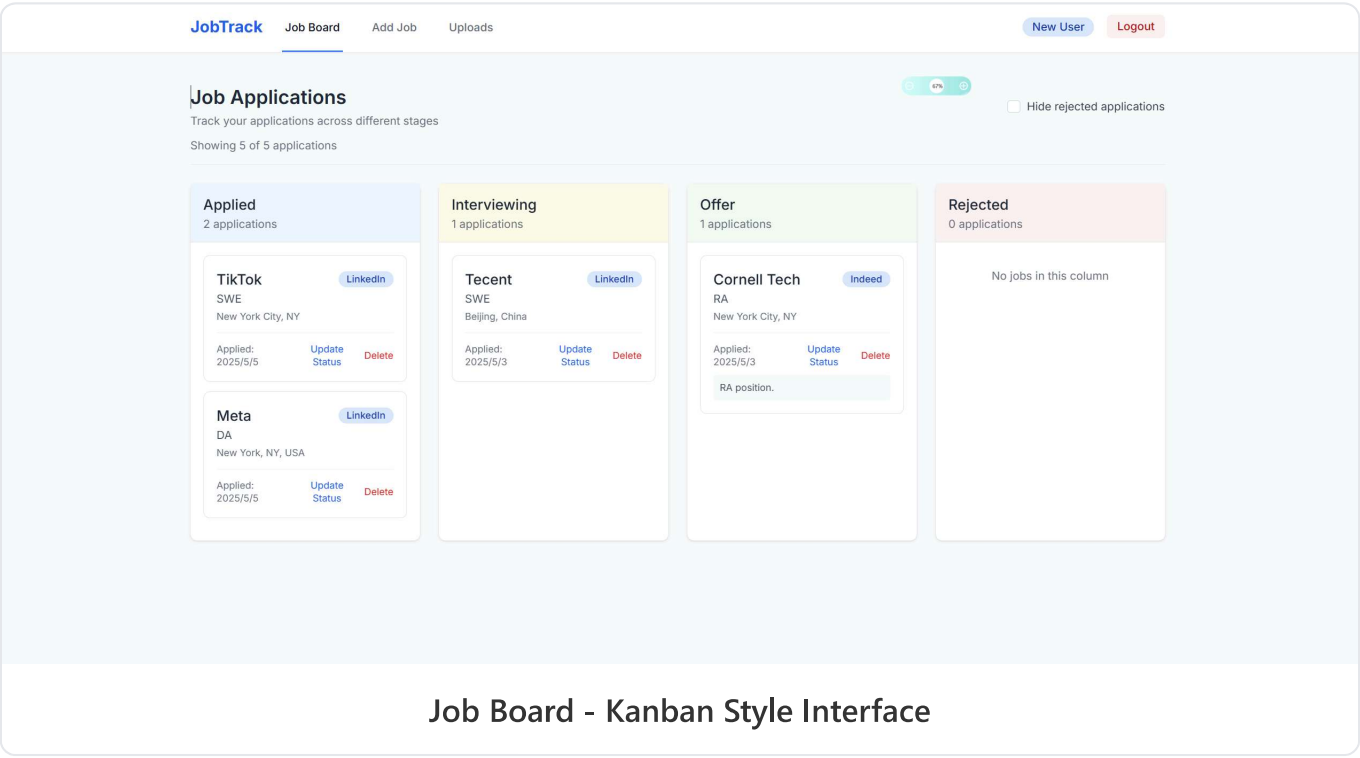
## Prototypes

---

The following screenshots showcase the key interfaces of the JobTrack application:



Login Page



Job Board - Kanban Style Interface

JobTrack

Job Board

Add Job

Uploads

New User

Logout

Add New Job Application

Track a new job you've applied for

Company Name

Position Title

Location

Date Applied

Application Source

Current Status

Notes (optional)

Cancel

Save

Add New Job Application Form

JobTrack

Job Board

Add Job

Uploads

New User

Logout

Uploaded Files

Manage your resumes, cover letters, and other job-related documents

Upload files or drag and drop

PDF, DOC, DOCX, PNG, JPG up to 10MB

No files uploaded yet

Document Upload Interface

## Starter Code

The project was built from scratch using Next.js as the primary framework. No specific starter templates were used beyond the initial Next.js project setup.

## Technology Choices

**Frontend:** The application uses Next.js for server-side rendering and API routes, with React for component-based UI development. Tailwind CSS was chosen for styling due to its utility-first approach and ease of creating responsive designs. The @hello-pangea/dnd library provides the drag-and-drop functionality for the job board interface.

**Backend:** MongoDB was selected as the database solution for its flexibility with document-based data structures and seamless integration with Next.js. NextAuth.js handles authentication and session management, providing a secure and straightforward way to implement user accounts.

**Deployment:** The application is deployed on Vercel, which offers excellent integration with Next.js applications and provides continuous deployment capabilities. MongoDB Atlas serves as the cloud-hosted database service.

## Learnings

---

### Challenges

1. **MongoDB Schema Design:** Designing the relationships between job applications and their status history required careful planning to ensure efficient queries and data integrity. Creating appropriate MongoDB models and ensuring proper references between collections was particularly challenging.
2. **Drag and Drop Implementation:** Implementing the drag-and-drop functionality for the job board was more complex than anticipated, especially ensuring that the backend properly updated when jobs were moved between columns. The status update functionality needed special attention to maintain data consistency.
3. **File Upload System:** Building a secure file upload system with proper validation and storage was challenging, particularly in ensuring files were properly associated with the correct user accounts. Managing file storage and retrieval while maintaining security was non-trivial.
4. **Authentication Integration:** Integrating NextAuth.js with MongoDB and ensuring proper user data isolation required careful implementation of middleware and session management. Ensuring that each user could only access their own data was critical for security.
5. **MongoDB Atlas Connectivity with Vercel:** Getting the deployed application to connect to MongoDB Atlas required configuration of the IP whitelist to allow Vercel's dynamic IP addresses, which was not immediately obvious.

## Unexpected Ease

1. **Vercel Deployment:** The deployment process with Vercel was surprisingly straightforward, with excellent integration for Next.js applications. The ability to automatically deploy on code pushes made the development workflow much smoother.
2. **Tailwind CSS:** Using Tailwind significantly accelerated the UI development process and made responsive design much easier than anticipated. The utility-first approach was particularly effective for building a consistent UI.
3. **Next.js API Routes:** The API route system in Next.js made it straightforward to create the necessary backend endpoints without setting up a separate server. The file-based routing structure was intuitive and helped organize the code effectively.

## Requirements Fulfillment

### Frontend/UI Met

**Minimum:** The HTML structure is semantic and well-organized, with responsive design ensuring content is accessible on all device sizes.

**Expectation:** The application maintains consistent styling throughout, with custom components that align with the overall design system.

**Next Level:** The application implements drag-and-drop UI patterns for job status management and custom modal components for status updates and timelines.

### Backend/Server Met

**Minimum:** The application is publicly accessible via HTTPS through Vercel.

**Expectation:** The backend is built with Next.js API routes that respond to frontend requests and centralize data in MongoDB.

**Next Level:** The server implements specialized logic for managing job status workflows and document storage.

## Authentication Met

**Minimum:** Users can persist their job data securely in the application.

**Expectation:** Each user has their own account with access limited to their own data.

## Deployment/DevOps Met

**Minimum:** The project uses continuous integration through Vercel's build system.

**Expectation:** The project uses continuous delivery, automatically deploying when code is pushed to the repository.

## Future Enhancements

---

If development were to continue, potential enhancements could include:

1. Supporting change and forgot password functionality for better account management
2. Integration with job boards like LinkedIn or Indeed
3. Analytics dashboard showing application statistics
4. Track file modification history (different versions of resumes and other documents)
5. Supporting search functionality for jobs applied to, enabling users to quickly find specific applications

## Conclusion

---

JobTrack successfully meets all the minimum and expected requirements for this project. The application demonstrates practical application of the technologies covered in class, creating a

useful tool that addresses a real-world need. The project effectively combines frontend, backend, authentication, and deployment techniques to create a cohesive full-stack application.